1. **System requirements**: Document specifications of the system being used for React Native development. This includes the CPU, RAM size, and Windows version.

**Device name** Zainul\_Laptop

**Processor** 13th Gen Intel(R) Core(TM) i7-13700HX 2.10 GHz

**Installed RAM** 16.0 GB (15.7 GB usable)

**Device ID**  0D2447CE-EA56-4EB0-8FB8-84F176C5B5ED

**Product ID**  00342-21098-15013-AAOEM

**System type**  64-bit operating system, x64-based processor

**Pen and touch** No pen or touch input is available for this display  
  
**Edition**  Windows 11 Home

**Version**  22H2

**Installed on**  ‎2023-‎10-‎20

**OS build** 22621.2506

**Serial number** PF3SHX50

**Experience** Windows Feature Experience Pack 1000.22677.1000.0

1. **Installation instructions**: Include step-by-step instructions for installing the necessary tools and dependencies required for the framework, such as Node.js, and the React Native CLI.

Downloading and Installing React Native/Android Studio

1. First go to the react native website <https://reactnative.dev>
2. Click on the “Development” tab
3. Select the “Guides” option from the drop down menu
4. Then click on “Environment setup” from the left hand side menu
5. Select “Setting up the development environment” from the drop down menu
6. Click on “React Native CLI Quickstart tab”
7. Choose your Development OS (which in our case for this course is Windows OS)
8. Select the Target OS (which in our case for this course is Android OS)
9. Install Node.js from <https://nodejs.org/en>
10. Download the LTS version
11. Go through the setup prompt and accept the “Terms in the License Agreement”.
12. Click next and leave everything as default from this point and make sure to not choose “Chocolatey”
13. Download Android Studio (in our case from Brightspace)
14. Double click on the downloaded setup process. (Android Studio may take a while to start up so do not keep clicking)
15. On the first screen of the setup prompt, click “Next”
16. On the second screen leave everything as default and click “Next”
17. On the third screen leave the installation location as it is and click “Next”
18. On the fourth screen, leave everything as default again and click “Install”
19. Once Android Studio has installed, select the checkbox which reads “Start Android Studio” click “Finish”.
20. A prompt will show up about sharing your usage statistics with Google. Either allow or deny access to share your usage statistics with Google.
21. A window will appear named “Android Studio Setup Wizard”. On this screen click “Next”.
22. On the second screen leave the option “Standard” selected and click “Next”
23. On the third screen select your preferred UI theme and click “Next”
24. On the fourth screen make sure you have the “Android SDK”, “Android SDK Platform”, “Android API” and “Android Virtual Device” selected. After this, select the directory you want this program to be installed in before clicking “Next”
25. On the fifth screen click “Next”
26. On the sixth screen read each “License Agreement” and click “Accept” for each. Then click “Finish”
27. Wait for Android Studio to install
28. Once Android Studio has installed click “Finish” and a window will open up to start or import a project
29. At the center of the page, click on the “More Actions” link
30. Then select “SDK Manager” from the drop down menu (a screen will show up)
31. Click on “Languages & Frameworks” and select “Android SDK” from the drop down menu”
32. Make sure that “Android API” is selected under “SDK Platforms”. Then click on “Show Package Details” and check to see if “Android SDK Platform 34” is selected under the “Android API 34” tab.
33. Select the “SDK Tools” tab and check to see if “Android SDK Build-Tools 34”, “Android Emulator”, “Intel x86 Emulator Accelerator (HAXM installer) - Deprecated” are selected
34. Click “Ok”
35. Right click on the Windows Icon on the taskbar
36. Select “Settings” from the list
37. Search “environment” and select either of the two options from the drop down menu
38. Click on the “New” button to create a new environment variable for the user
39. Type “ANDROID\_HOME” into the “Variable Name” box
40. Find out where you installed the SDK to write it as the “Variable Value” by opening up the Android Studio window again. Then, click on “More Actions” and select “SDK Manager”. At the top of the screen in the “Android SDK Location” box you will find the location.
41. Now copy that location and paste it into the “Variable Value” box and click “OK”
42. Now we need to add in the platform tools to the path variable
43. To add the platform tools to the path variable, select the path variable from the same “Environment Variables” window.
44. Now click “Edit”
45. Click on “Browse” and find out where you stored the Android SDK folder
46. Under the Android SDK folder select the “platform tools” folder and click “OK”
47. Click “OK” again to exit the “Environment Variables” window
48. Now your Android Studio should be configured so that a React Native project can be built

—-------------------------------------------------------------------------------------------------------------------------

Creating/Starting a React Native Project

1. To Start or create a React Native project, we need to select a location to create a folder for our projects, outside of the OneDrive folder and Users Directory.
2. After selecting the location open that location in command prompt
3. Then, enter this command “npx react-native@latest init Awesome Project” and press enter. (Note: Awesome project can be replaced with whatever you want to call your project)
4. Enter “y” when asked “Ok to proceed?”
5. If you get a message saying that your npm is out of date, type “npm i -g npm” and enter.
6. Now open Visual Studio Code
7. Select the “File Tab” and then “Open Folder” from the drop down menu.
8. Double click on the folder where your project was created and click on “Select Folder”
9. If you get a prompt asking if you trust the author, select “Yes, I trust the author.”
10. The important files (in our case) are “app.json”, “App.tsx”, “index.js”, “metro.config.js” and the “package.json” file.
11. If you are not comfortable using typescript, change it to javascript rename “App.tsx” to “App.jsx”.
12. Upon completing this step, there will be red lines in the “App.jsx” file so we have to simply delete them to resolve the issue and make your file a javascript extended file.

—-------------------------------------------------------------------------------------------------------------------------  
Device Emulator Setup

1. Before we can test and run our app, we need to have a virtual device setup. We need to create an emulator or virtual machine image.
2. Open Android Studio
3. Click on ‘More Actions”
4. Select “Virtual Device Manager” from the drop down menu
5. Then click on “Create Device”
6. For our case, we will select the “Pixel 7” from the device list and click “Next”
7. On the second screen, by clicking on the download button, download “R”. Once downloaded, click “Finish”
8. Select “Next”
9. On the third screen, select “Portrait” and click “Finish”
10. Now click on the start button to start your device
11. If you get an “adb” Windows Firewall Prompt, click “Allow Access”. This will allow you to run your app on your phone as well.
12. The Android Emulator will take a while to start. If you get a black screen just wait. However, if the black screen does not go away click on the start button to ensure that the device is on.
13. Now you should have an Android Device Emulator

—-------------------------------------------------------------------------------------------------------------------------

Run Project in Android Studio

1. To get your project to start the first time, open it through android studio.
2. Click on the “Open” button in the middle of the screen and open the “android” folder where your project was created.
3. Another screen will now open and it may take a little while to load everything properly
4. After everything is loaded and installed, if you get a prompt to update the project just ignore it and close it.
5. Select the “Build” tab and choose “Make Project”. Wait for the app to build.
6. Once you see “Build Successful” click on the “Run” tab and select “Run” from the drop down menu to run your app.
7. Once your app opens, you should see an error reading “Unable to Load Script”
8. To fix this open the directory in command prompt where the app was created and enter “npm run start”
9. You should see a list of options out of which you should just type “r” in our case
10. Now your app should be running in the device emulator
11. To test if the correct program is running, go into your App.jsx folder and change something like text. If that text changes when you look at your app again, it means that you are running the correct live version.

—-------------------------------------------------------------------------------------------------------------------------

Running Your React Native Project from the Command Line

1. To run the app from the command line, navigate to the directory where your app was created and open it in command prompt
2. Then type “npm run start” and press enter
3. Then press “a” (in our case) from the list of given options. It can take a little while to fully build so be patient.
4. Once everything is complete, your app should be running successfully using the command line.

—-------------------------------------------------------------------------------------------------------------------------

Z Build Error

1. If you are trying to run your react native project from the command line and you are getting the error: “No matching variant of project :gradle-plugin found”, it means that there are multiple JDK’s installed. To fix this, follow the steps below
2. Navigate to “environment variables” by searching it in the windows search bar and click on the first option that comes up
3. Then in the popup, click the “Environment Variables” button
4. Then click on path, and select “Edit”
5. Check if there are any additional JDK paths there
6. Then go under system variables, click on path, and then the “Edit” button
7. Check if there are any additional JDK paths there and delete them by pressing the “Delete” button.
8. Next, add the path for the Android Studio JDK which is installed in the program files directory, under android, under android studio, under the jdr folde, and finally under the bin folder..
9. Copy the path
10. Go back to the environment variables window, click “New” and paste the path into the box
11. Then click “Ok” and close the window
12. Then open up a fresh cmd window or, if that doesn't work, restart your computer.

—-------------------------------------------------------------------------------------------------------------------------

Z - React Native Doctor

1. A tool that can be used to fix your React Native Development Environment is “React Native Doctor”. (Note: Only works if your code is working and if the issue is being caused by the React Native installation process.
2. To run it, go to the directory where your project was created and open it in cmd
3. Enter “npx react-native doctor”
4. Check to see if there are any issues. If there are, press “f” and it will automatically try to resolve those issues.
5. **Configuration steps**: Detail any necessary configuration steps required to set up the framework, such as setting environment variables or configuring project settings.
6. At the center of the page, click on the “More Actions” link
7. Then select “SDK Manager” from the drop down menu (a screen will show up)
8. Click on “Languages & Frameworks” and select “Android SDK” from the drop down menu”
9. Make sure that “Android API” is selected under “SDK Platforms”. Then click on “Show Package Details” and check to see if “Android SDK Platform 34” is selected under the “Android API 34” tab.
10. Select the “SDK Tools” tab and check to see if “Android SDK Build-Tools 34”, “Android Emulator”, “Intel x86 Emulator Accelerator (HAXM installer) - Deprecated” are selected
11. Click “Ok”
12. Right click on the Windows Icon on the taskbar
13. Select “Settings” from the list
14. Search “environment” and select either of the two options from the drop down menu
15. Click on the “New” button to create a new environment variable for the user
16. Type “ANDROID\_HOME” into the “Variable Name” box
17. Find out where you installed the SDK to write it as the “Variable Value” by opening up the Android Studio window again. Then, click on “More Actions” and select “SDK Manager”. At the top of the screen in the “Android SDK Location” box you will find the location.
18. Now copy that location and paste it into the “Variable Value” box and click “OK”
19. Now we need to add in the platform tools to the path variable
20. To add the platform tools to the path variable, select the path variable from the same “Environment Variables” window.
21. Now click “Edit”
22. Click on “Browse” and find out where you stored the Android SDK folder
23. Under the Android SDK folder select the “platform tools” folder and click “OK”
24. Click “OK” again to exit the “Environment Variables” window
25. Now your Android Studio should be configured so that a React Native project can be built

—-------------------------------------------------------------------------------------------------------------------------Device Emulator Setup

1. Before we can test and run our app, we need to have a virtual device setup. We need to create an emulator or virtual machine image.
2. Open Android Studio
3. Click on ‘More Actions”
4. Select “Virtual Device Manager” from the drop down menu
5. Then click on “Create Device”
6. For our case, we will select the “Pixel 7” from the device list and click “Next”
7. On the second screen, by clicking on the download button, download “R”. Once downloaded, click “Finish”
8. Select “Next”
9. On the third screen, select “Portrait” and click “Finish”
10. Now click on the start button to start your device
11. If you get an “adb” Windows Firewall Prompt, click “Allow Access”. This will allow you to run your app on your phone as well.
12. The Android Emulator will take a while to start. If you get a black screen just wait. However, if the black screen does not go away click on the start button to ensure that the device is on.

Now you should have an Android Device Emulator

**4. Project creation**: Outline the steps involved in creating a new project using the framework, including any necessary setup or configuration.  
  
Creating/Starting a React Native Project

1. To Start or create a React Native project, we need to select a location to create a folder for our projects, outside of the OneDrive folder and Users Directory.
2. After selecting the location open that location in command prompt
3. Then, enter this command “npx react-native@latest init Awesome Project” and press enter. (Note: Awesome project can be replaced with whatever you want to call your project)
4. Enter “y” when asked “Ok to proceed?”
5. If you get a message saying that your npm is out of date, type “npm i -g npm” and enter.
6. Now open Visual Studio Code
7. Select the “File Tab” and then “Open Folder” from the drop down menu.
8. Double click on the folder where your project was created and click on “Select Folder”
9. If you get a prompt asking if you trust the author, select “Yes, I trust the author.”
10. The important files (in our case) are “app.json”, “App.tsx”, “index.js”, “metro.config.js” and the “package.json” file.
11. If you are not comfortable using typescript, change it to javascript rename “App.tsx” to “App.jsx”.
12. Upon completing this step, there will be red lines in the “App.jsx” file so we have to simply delete them to resolve the issue and make your file a javascript extended file.
13. **Running the project**: Detail how to run the project in an Android Device Simulator.  
      
    Device Emulator Setup
14. Before we can test and run our app, we need to have a virtual device setup. We need to create an emulator or virtual machine image.
15. Open Android Studio
16. Click on ‘More Actions”
17. Select “Virtual Device Manager” from the drop down menu
18. Then click on “Create Device”
19. For our case, we will select the “Pixel 7” from the device list and click “Next”
20. On the second screen, by clicking on the download button, download “R”. Once downloaded, click “Finish”
21. Select “Next”
22. On the third screen, select “Portrait” and click “Finish”
23. Now click on the start button to start your device
24. If you get an “adb” Windows Firewall Prompt, click “Allow Access”. This will allow you to run your app on your phone as well.
25. The Android Emulator will take a while to start. If you get a black screen just wait. However, if the black screen does not go away click on the start button to ensure that the device is on.
26. Now you should have an Android Device Emulator

—-------------------------------------------------------------------------------------------------------------------------

Run Project in Android Studio

1. To get your project to start the first time, open it through android studio.
2. Click on the “Open” button in the middle of the screen and open the “android” folder where your project was created.
3. Another screen will now open and it may take a little while to load everything properly
4. After everything is loaded and installed, if you get a prompt to update the project just ignore it and close it.
5. Select the “Build” tab and choose “Make Project”. Wait for the app to build.
6. Once you see “Build Successful” click on the “Run” tab and select “Run” from the drop down menu to run your app.
7. Once your app opens, you should see an error reading “Unable to Load Script”
8. To fix this open the directory in command prompt where the app was created and enter “npm run start”
9. You should see a list of options out of which you should just type “r” in our case
10. Now your app should be running in the device emulator
11. To test if the correct program is running, go into your App.jsx folder and change something like text. If that text changes when you look at your app again, it means that you are running the correct live version.

**6. Troubleshooting**: Include information on how to troubleshoot common issues that may arise during setup or development, such as debugging and error messages. This may need to be updated in the future.  
  
Z Build Error

1. If you are trying to run your react native project from the command line and you are getting the error: “No matching variant of project :gradle-plugin found”, it means that there are multiple JDK’s installed. To fix this, follow the steps below
2. Navigate to “environment variables” by searching it in the windows search bar and click on the first option that comes up
3. Then in the popup, click the “Environment Variables” button
4. Then click on path, and select “Edit”
5. Check if there are any additional JDK paths there
6. Then go under system variables, click on path, and then the “Edit” button
7. Check if there are any additional JDK paths there and delete them by pressing the “Delete” button.
8. Next, add the path for the Android Studio JDK which is installed in the program files directory, under android, under android studio, under the jdr folde, and finally under the bin folder..
9. Copy the path
10. Go back to the environment variables window, click “New” and paste the path into the box
11. Then click “Ok” and close the window
12. Then open up a fresh cmd window or, if that doesn't work, restart your computer.

—-------------------------------------------------------------------------------------------------------------------------

Z - React Native Doctor

1. A tool that can be used to fix your React Native Development Environment is “React Native Doctor”. (Note: Only works if your code is working and if the issue is being caused by the React Native installation process.
2. To run it, go to the directory where your project was created and open it in cmd
3. Enter “npx react-native doctor”
4. Check to see if there are any issues. If there are, press “f” and it will automatically try to resolve those issues.

**7. Resources**: Provide links to additional resources and documentation that can be used for reference and further learning, such as official documentation, tutorials, and Stack Overflow answers.

1. \*\*React Native Official Documentation\*\*: This is the best place to start, offering comprehensive guides, tutorials, and API references.

- [React Native Official Documentation](https://reactnative.dev/docs/getting-started)

2. \*\*React Native GitHub Repository\*\*: Useful for understanding the latest updates, issues, and contributions.

- [React Native GitHub](https://github.com/facebook/react-native)

3. \*\*React Native Tutorial for Beginners\*\*: A step-by-step guide for beginners.

- [React Native Tutorial - Programming with Mosh](https://programmingwithmosh.com/react-native/react-native-tutorial-for-beginners/)

4. \*\*React Navigation\*\*: Learn about navigation in React Native.

- [React Navigation Documentation](https://reactnavigation.org/)

5. \*\*React Native Community\*\*: Engage with the React Native community for support and discussions.

- [React Native Community](https://reactnative.dev/community)

6. \*\*Stack Overflow\*\*: A great resource for specific questions and troubleshooting. You can find a wealth of information and solutions to common problems here.

- [React Native Questions - Stack Overflow](https://stackoverflow.com/questions/tagged/react-native)

7. \*\*React Native Blog\*\*: Follow the official React Native blog for the latest news and updates.

- [React Native Blog](https://reactnative.dev/blog)

8. \*\*YouTube Tutorials\*\*: There are many video tutorials available on YouTube that can be helpful for visual learners.

- [React Native Crash Course](https://www.youtube.com/watch?v=Hf4MJH0jDb4)

9. \*\*Medium Articles on React Native\*\*: Medium hosts numerous articles and stories about React Native, from basic tutorials to advanced topics.

- [React Native on Medium](https://medium.com/tag/react-native)

10. \*\*React Native Course on Udemy\*\*: Udemy offers comprehensive courses for all levels, often with practical projects.

- [React Native - The Practical Guide](https://www.udemy.com/course/react-native-the-practical-guide/)